REMARKS

Applicants respectfully request consideration of the subject application. Applicants amended claims 1, 6, 11, 16, 21, 25, and 30 the limitations already present in the claims, thus Applicants added no new matter through the amendments.

Claim Rejections under 35 U.S.C. § 103(a)

Claims 1-15 and 25-34 have been rejected under 35 U.S.C. § 103(a) as being unpatentable over Blaker, et al. U.S. Publication No. 2002/0004904 ("Blaker") in view of U.S. Patent No. 6,584,567 to Bellwood ("Bellwood") and in view of U.S. Publication No. 2002/0146128 to Mauro et al ("Mauro").

Claims 16-18, 20-22, and 24 have been rejected under 35 U.S.C. § 103(a) as being unpatentable over Blaker in view of Mauro.

Claims 19 and 23 have been rejected under 35 U.S.C. § 103(a) as being unpatentable over Blaker in view of Mauro and further in view of U.S. Patent No. 5,925,123 to Tremblay ("Tremblay").

Claims 1-15 and 25-34

Claim 1

Claim1 requires a single macro operation representing a plurality of primitive security operations wherein a single execution unit of a second processor having a plurality of execution units performs the plurality of primitive security operations.

Specifically, claim 1 requires calling a single macro instruction operation representing a plurality of primitive security operations. Furthermore, claim 1 requires executing the plurality of primitive security operations at a second processor having a plurality of execution units that each can perform the single macro instruction operation, wherein a single execution unit of said plurality of execution units performs the plurality of primitive security operations that correspond to the single macro instruction. Moreover, claim 1 requires the single macro instruction selected from a group of macro instructions including a first key exchange operation macro, a second key exchange operation macro, a finish operation macro, and a server full handshake operation macro, wherein the first key exchange macro operation represents a primitive security operation including a decrypt operation, a group of modular arithmetic operations, and seventy-eight hash operations, the second key exchange operation macro represents a primitive security operation including one decrypt operation, a group of modular arithmetic operations, and twenty-two hash operations, the finish operation macro represents a primitive security operation including one decrypt operation, an encrypt operation, twelve hash operations, and the server full handshake operation macro represents a primitive security operation including a decrypt operation, two encrypt operations, a set of modular arithmetic operations, and thirty-five hash operations.

The combination describes a proxy participating in secure communications between the client and a first server according to Bellwood[1]. The proxy, the client,

[1] Bellwood describes a method of enabling a proxy to participate in secure communication between a client and a set of servers. (Bellwood, Abstract). A first secure connection is established between the client and the proxy. (Bellwood, Abstract). The method also describes the proxy participating in secure communications between the client and a first server. (Bellwood, Abstract).

and the first server have a host processor and a cryptographic accelerator processor according to Blaker[2]. A command block created by a host processor is stored in memory where a cryptographic accelerator processor accesses and executes the single instruction that references a single operand according to Blaker. Furthermore, the cryptographic accelerator processor receives a single command (i.e. command ID) to execute a single primitive cryptographic function according to Mauro[3]. In order to run the single primitive cryptographic function the cryptographic accelerator must load an image file containing the executable instructions "required to execute the particular primitive function." (Mauro, para. [0029]).

The combination fails to describe calling with a single macro instruction operation representing a plurality of primitive security operations. Furthermore, the

[2] Blaker describes a command block created by a host processor. The command block includes commands for execution, locations for storing, and parameters to be loaded by the cryptographic accelerator processor. Blaker describes a one to one relationship between operand and instruction. For instance, paragraph 11 describes, "one or more operands are downloaded into the local memory from the system memory and the cryptographic processor executes *an instruction* that references *one* of the downloaded operands." (Blaker, para. 11, emphasis added). Thus, for every operand in the command block the cryptographic accelerator processor executes one instruction and the execution units described by Blaker only process one type of operand.

[3] Mauro describes a DSP performing a primitive cryptographic function when the CPU sends a command and/or many executable instructions to execute one primitive cryptographic function on the DSP. Specifically, Mauro describes "when a primitive cryptographic function is required, the CPU downloads the DSP assembly image into the DSP" and sends a command to the DSP to execute the primitive cryptographic function. (Mauro, para. [0029]). This image contains the DSP executable instructions (microcode) "required to execute the particular primitive cryptographic function." (Mauro, para. [0029]). The CPU then sends a command to the DSP to execute the particular primitive function. (Mauro, para. [0029]). Mauro also describes the above one command to execute one primitive cryptographic function for the following primitive cryptographic functions: hash (para. [0033]), modular exponentiation (paras. [0031], [0037]; Figure 3), encryption/decryption (paras. [0034], [0040], [0041]; Figure 5; Figure 6), and modular math (para. [0039]; Figure 4). Thus, the CPU sends one command and/or executable instructions to perform one primitive cryptographic function.

For example, Mauro describes using the DSP to accelerate the modular exponentiation used in Diffie-Hellman and RSA key exchange algorithms. (Mauro, paras. [0031], [0037]). The CPU downloads the executable instructions that cause the DSP to execute the exponentiation function which is a primitive cryptographic function. (Mauro, para. [0031], [0037]). Thus, Mauro describes using a command and/or many executable instructions to perform one primitive cryptographic function.

combination fails to describe a single macro instruction operation selected from a group of macro operations including a first key exchange macro operation, a second key exchange macro operation, a finish macro operation, a server full handshake macro operation, and the specific plurality of primitive security operations listed in claim 1.

Moreover, the combination fails to describe or suggest a second processor having a plurality of execution units that each can perform the single macro instruction operation, wherein a single execution unit of said plurality of execution units performs the plurality of primitive security operations that correspond to the single macro instruction. The combination, as discussed above, describes execution units that only perform one primitive corresponding to the operand or command ID sent to the cryptographic accelerator processor. Furthermore, since the combination only describes executing one primitive for any one operand[4] or command ID[5], as discussed above, the combination fails to describe one of the plurality of execution units to perform the plurality of primitive security operations corresponding to the macro security operation.

Thus, the combination fails to render claim 1 obvious because the combination fails to describe or suggest all the limitations of claim 1.

[4] As discussed above, Blaker describes a one to one relationship between operand to instruction, thus fails to describe or suggest a second processor to perform the *plurality* of primitive security operations in response to the macro security operation from said first processor. Therefore the execution units described in Blaker only perform one primitive corresponding to one operand.

[5] Mauro describes the command sent to a DSP includes the primitives such as encrypt, decrypt, and hash. (Mauro, para. [0038]). Moreover, Mauro describes a CPU that sends a command and/or executable instructions to perform one primitive cryptographic function. Therefore, Mauro fails to describe or suggest to perform the plurality of primitive security operations in response to the macro security operation from said first processor, as discussed above.

Claims 2-5

Applicants respectfully submit that claims 2-5 are dependent on claim 1; therefore, claims 2-5 include the same limitations as claim 1. As such, claims 2-5 are allowable for at least the same reasons as claim 1.

Claim 6

Claim 6 requires calling a single macro security operation representing a set of primitive security operations, wherein the single macro security operation is a server full handshake macro operation. Moreover, claim 6 requires the set of primitive security operations comprising, generating a secret and a key material, creating a first finished hash for a client message, creating a second finished hash for a server message, and creating a finished message.

The combination as discussed above fails to describe the above limitations of claim 6 because the combination fails to describe server full handshake macro security operation that causes a second processor to perform all of the set of primitive security operations required by claim 6.

Claims 7-10

Applicants respectfully submit that claims 7-10 are dependent on claim 6; therefore, claims 7-10 include the same limitations as claim 6. As such, claims 7-10 are allowable for at least the same reasons as claim 6.

Claim 30

Claim 30 requires similar limitations[6] as claim 6, thus is not rendered obvious by the combination for at least similar reasons as discussed for the claim 6 limitations that are similar to claim 30.

Claims 31-34

Applicants respectfully submit that claims 31-34 are dependent on claim 30; therefore, claims 31-34 include the same limitations as claim 30. As such, claims 31-34 are allowable for at least the same reasons as claim 30.

Claim 11

Claim 11 requires the second network element to call a macro security operation associated with a plurality of primitive security operations. Furthermore, claim 11 requires the second network element to execute the plurality of primitive security operations at a second processor in response to the macro security operation selected from a group of macro operations. The group of macro operations including a first key exchange macro operation, a second key exchange macro operation, a finish macro operation, and a server full handshake macro operation, wherein the first key exchange macro operation associated with the plurality of primitive security operations including a decrypt operation, a group of

[6] Claim 30 requires calling a single macro security operation associated with a set of primitive security operations, wherein the single macro instruction is a server full handshake macro instruction. Furthermore, claim 30 requires performing the set of primitive security operations at a second one of the set of processors in response to the single macro security operation, the set of primitive security operations comprising, generating a secret and a key material, creating a first finished hash for a client message, creating a second finished hash for a server message, and creating a finished message.

modular arithmetic operations, and seventy-eight hash operations, the second key exchange macro operation associated with the plurality of primitive security operations including one decrypt operation, a group of modular arithmetic operations, and twenty-two hash operations, the finish macro operation associated with the plurality of primitive security operations including one decrypt operation, an encrypt operation, twelve hash operations, and the server full handshake macro operation associated with the plurality of primitive security operations including a decrypt operation, two encrypt operations, a set of modular arithmetic operations, and thirty-five hash operations.

The combination, as discussed above, fails to describe a macro operation associated with a plurality of primitive security operations. Specifically, the combination fails to describe macro operations including a first key exchange macro operation, a second key exchange macro operation, a finish macro operation, and a server full handshake macro operation associated with the primitive security operations as required in claim 11. Therefore, the combination fails to render claim 11 obvious.

Claims 12-15

Applicants respectfully submit that claims 12-15 are dependent on claim 11; therefore, claims 12-15 include the same limitations as claim 11. As such, claims 12-15 are allowable for at least the same reasons as claim 11.

Claim 25

Claim 25 requires similar limitations[7] as claim 11, thus is not rendered obvious by the combination for at least similar reasons as discussed for the claim 11 limitations that are similar to claim 25.

Claims 26-29

Applicants respectfully submit that claims 26-29 are dependent on claim 25; therefore, claims 26-29 include the same limitations as claim 25. As such, claims 26-29 are allowable for at least the same reasons as claim 25.

Claims 16-24

Claim 16

Claim 16 requires a first processor to call a macro security operation associated with a plurality of primitive security operations to establish a secure session. Furthermore, claim 16 require the macro security operation selected from a group of macro security operations including a first key exchange macro security operation, a second key exchange macro security operation, a finish macro security operation, and a server full handshake macro security operation. Moreover, Claim 6

[7] Claim 25 requires executing a macro security operation associated with a plurality of primitive security operations. Moreover, claim 25 requires the macro security operation selected from a group including a key exchange macro, a finish macro, and a server full handshake macro, wherein the key exchange macro associated with the plurality of primitive security operations including one decrypt operation, a group of modular arithmetic operations, and twenty-two hash operations, the key exchange operation macro associated with the plurality of primitive security operations including a decrypt operation, a group of modular arithmetic operations, and seventy-eight hash operations, the finish macro associated with the plurality of primitive security operations including one decrypt operation, an encrypt operation, twelve hash operations, and the server full handshake macro associated with the plurality of primitive security operations including a decrypt operation, two encrypt operations, a set of modular arithmetic operations.

requires the first key exchange operation associated with a decrypt operation, a group of modular arithmetic operations, and seventy-eight hash operations, the second key exchange operation associated with one decrypt operation, a group of modular arithmetic operations, and twenty-two hash operations, the finish operation macro represents one decrypt operation, an encrypt operation, twelve hash operations, and the server full handshake operation macro associated with a decrypt operation, two encrypt operations, a set of modular arithmetic operations, and thirty-five hash operations.

The combination describes a host processor and a cryptographic accelerator processor according to Blaker[8]. A command block created by a host processor is stored in memory where a cryptographic accelerator processor accesses and executes the single instruction that references a single operand according to Blaker. Furthermore, the cryptographic accelerator processor receives a single command to execute a single primitive cryptographic function according to Mauro[9]. In order to run the single primitive cryptographic function the cryptographic accelerator must load an image file containing the executable instructions "required to execute the particular primitive function." (Mauro, para. [0029]).

The combination, as discussed above, fails to describe a macro operation associated with a plurality of primitive security operations. Specifically, the combination fails to describe macro operations including a first key exchange macro

[8] As discussed above, Blaker describes a one to one relationship between operand and instruction, thus fails to describe or suggest a second processor to perform the *plurality* of primitive security operations in response to the macro security operation from said first processor.

[9] Mauro, discussed above, describes a CPU that sends a command and/or executable instructions to perform one primitive cryptographic function. Therefore, Mauro fails to describe or suggest to perform the plurality of primitive security operations in response to the macro security operation from said first processor.

operation, a second key exchange macro operation, a finish macro operation, and a server full handshake macro operation associated with the primitive security operations as required in claim 6. Therefore, the combination fails to render claim 6 obvious.

Claims 17-20

Applicants respectfully submit that claims 17-20 are dependent on claim 16; therefore, claims 17-20 include the same limitations as claim 16. As such, claims 17-20 are allowable for at least the same reasons as claim 16.

Claim 21

Claim 21 requires similar limitations[10] as claim 16, thus is not rendered obvious by the combination for at least similar reasons as discussed for the claim 16 limitations that are similar to claim 21.

Moreover, Claim 21 requires a plurality of execution units coupled to the request unit, one of the plurality of execution units to perform the plurality of primitive security operations corresponding to the macro security operation. The combination, as discussed above, describes execution units that only perform one

[10] Claim 21 requires a first processor to give the command for a macro security operation associated with a plurality of primitive security operations, the macro security operation selected from a group including a key exchange operation macro, a finish operation macro, and a server full handshake operation macro, wherein the key exchange operation macro associated with the plurality of primitive security operations including one decrypt operation, a group of modular arithmetic operations, and twenty-two hash operations, the key exchange operation macro associated with the plurality of primitive security operations including a decrypt operation, a group of modular arithmetic operations, and seventy-eight hash operations, the finish operation macro associated with the plurality of primitive security operations including one decrypt operation, an encrypt operation, twelve hash operations, and the server full handshake operation macro associated with the plurality of primitive security operations including a decrypt operation, two encrypt operations, a set of modular arithmetic operations, and thirty-five hash operations.

primitive corresponding to the operand or command ID sent to the cryptographic accelerator processor. Furthermore, since the combination only describes executing one primitive for any one operand[11] or command ID[12], as discussed above, the combination fails to describe one of the plurality of execution units to perform the plurality of primitive security operations corresponding to the macro security operation.

For at least the above reasons, the combination fails to render obvious claim 21.

Claims 22-24

Applicants respectfully submit that claims 22-24 are dependent on claim 21; therefore, claims 22-24 include the same limitations as claim 21. As such, claims 22-24 are allowable for at least the same reasons as claim 21.

[11] As discussed above, Blaker describes a one to one relationship between operand and instruction, thus fails to describe or suggest a second processor to perform the *plurality* of primitive security operations in response to the macro security operation from said first processor. Therefore, the execution units described in Blaker only perform one primitive corresponding to one operand.

[12] Mauro describes the command sent to a DSP includes the primitives such as encrypt, decrypt, and hash. (Mauro, para. [0038]). Moreover, Mauro describes a CPU that sends a command and/or executable instructions to perform one primitive cryptographic function. Therefore, Mauro fails to describe or suggest to perform the plurality of primitive security operations in response to the macro security operation from said first processor, as discussed above.

Conclusion

If the allowance of these claims could be facilitated by a telephone conference, the Examiner is invited to contact Daniel De Vos at (408) 720-8300. If there are any additional charges, please charge our Deposit Account No. 02–2666.

Respectfully submitted,
BLAKELY, SOKOLOFF, TAYLOR & ZAFMAN LLP

Dated: 11/6 , 2006

Thomas C. Webster
Registration No. 46,154

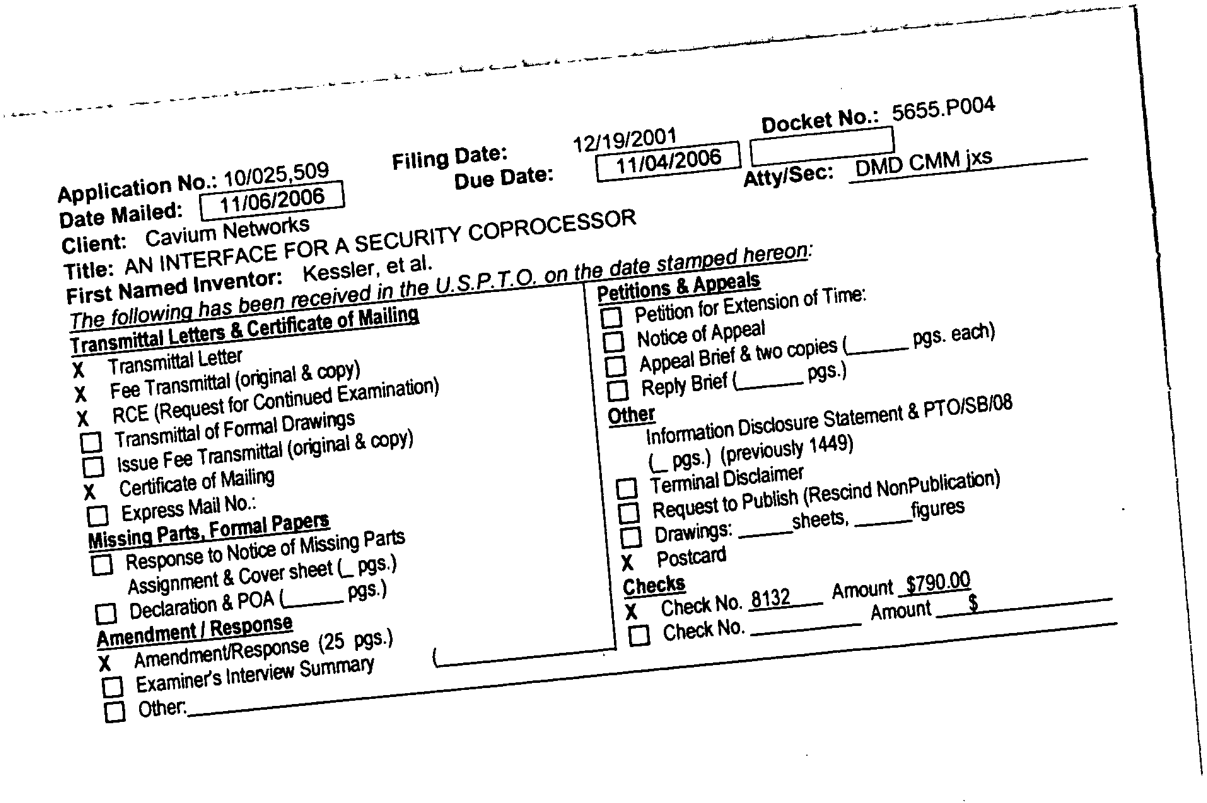
Customer No. 08791
12400 Wilshire Boulevard
Seventh Floor
Los Angeles, CA 90025-1030
(408) 720-8300
Fax (408) 720-8383

FIRST CLASS CERTIFICATE OF MAILING

I hereby certify that this correspondence is being deposited with the United States Postal Service as first class mail on the date indicated below with sufficient postage addressed to:

Mail Stop RCE
Commissioner For Patents
P.O. Box 1450
Alexandria VA 22313-1450

11/06/2006
Judy L. Steinkraus



Application No.: 10/025,509 Filing Date: 12/19/2001 Docket No.: 5655.P004
Date Mailed: 11/06/2006 Due Date: 11/04/2006 Atty/Sec: DMD CMM jxs
Client: Cavium Networks
Title: AN INTERFACE FOR A SECURITY COPROCESSOR
First Named Inventor: Kessler, et al.
The following has been received in the U.S.P.T.O. on the date stamped hereon:

Transmittal Letters & Certificate of Mailing
X Transmittal Letter
X Fee Transmittal (original & copy)
X RCE (Request for Continued Examination)
□ Transmittal of Formal Drawings
□ Issue Fee Transmittal (original & copy)
X Certificate of Mailing
□ Express Mail No.:

Missing Parts, Formal Papers
□ Response to Notice of Missing Parts
Assignment & Cover sheet (_ pgs.)
□ Declaration & POA (_____ pgs.)

Amendment / Response
X Amendment/Response (25 pgs.)
□ Examiner's Interview Summary
□ Other:

Petitions & Appeals
□ Petition for Extension of Time:
□ Notice of Appeal
□ Appeal Brief & two copies (_____ pgs. each)
□ Reply Brief (_____ pgs.)

Other
Information Disclosure Statement & PTO/SB/08
(_ pgs.) (previously 1449)
□ Terminal Disclaimer
□ Request to Publish (Rescind NonPublication)
□ Drawings: _____sheets, _____figures
X Postcard

Checks
X Check No. 8132 Amount $790.00
□ Check No. _________ Amount ____$